Forensic Analysis of ODF 1.2 Deletion Change-Tracking Model

I set out to create a change-tracked deletion example for ODF 1.2 that illustrates the kind of general case that the current specification and its implementations handle.

However, I found that I needed to engage in a forensic analysis because the case I chose encountered an edge in the widely-used code base.

Intricate single-deletion cases are produced correctly, but not consumed properly. There can be silent corruption of the `<text:deletion>` element and inability to restore the original text.

The Interesting Case of Deletion Tracking

Deletion tracking is underspecified in the current ODF 1.0-1.2 specifications. However, it is quite successfully implemented, suggesting that the problem is in the specification failing to capture the necessary information from which to make independent interoperable implementations.

This note demonstrates the “ground truth” for current deletion tracking. It is offered to indicate that there is much that needs to be comprehended in the current ODF 1.x practice so that it can be preserved in any improvement of the specification and the cases that are covered.

Format of Change-Tracking Document Samples

The present document includes one tracked change. To observe the change tracking you will need to use an ODF 1.0-1.2 consumer that implements tracked changes. Although this document was produced with changes set to visible, you may need to adjusted settings in your software to see the changes. It is likely that pre-1.2 implementations will have difficulties.

For reference and comparison, there are also screen captures of various stages of the tracked-change material. In those samples, the following conventions apply:

- **Insertions** are shown with [green highlighting](#)
- **Deletions** are show with [red highlighting](#)
- **Font changes** shown with [double-underlining](#)
- **Comments** provide observations about the changes

CT-2011-04-04-1123-DeletionTrackingForensics-ODF1.2-Lo3.3.2-dh
With software in the OO.o code-base family, you can use option settings such as those above to view the changes in the document sample, below, in the same form as in the screen captures.

This document was produced using Libre Office 3.3.2 on Microsoft Windows 7. It is identified as an ODF 1.2 Document.

**The Tracked Deletion**
1. This is list-item-a
2. This is list-item-b
   - This is some text hanging below the list-item-b
3. This is list-item-c
   - This is text hanging below the list-item-c
4. This is a paragraph of text following the list.

**Many Slip Twixt Cup and Lip**
The list and text, above, has a change-tracked deletion. This may or may not be visible in the OpenDocument Consumer that you use to inspect this document.
If the change tracking is visible, it may or may not display as the tracked-deletion that I created. That is because this apparently-simple deletion across some list and non-list text reveals an edge case in all implementations that I have checked.

Here is the single deletion that I made and what I saw when I made it:

![Image](CT-2011-04-02-2223-TrackedDeletion-ODF1.2-Lo3.3.2-dh.png)

Figure 1: The expected change-marking for the text and deletion I made

What's important is that the original list only has 3 items, not 4. The last line is not in the list. (The hanging paragraphs that are immediately below the list items numbered 2 and 3 are additional paragraphs of those list items.)

If this change is rejected, the text should be as shown above but with no high-lighting.

If the change is accepted (or the showing of the change-marking is turned off), the result should be a simple two-item list:

![Image](CT-2011-04-03-1242-TrackedNoShowRegression-ODF1.2-Lo3.3.2-dh.png)

Figure 2: Result from Accepting/Hiding the Tracked Deletion

Note: This is screen capture CT-2011-04-02-2223-TrackedDeletion-ODF1.2-Lo3.3.2-dh.png made before saving this file as the .odt of the same name. Different consumer versions will present the annotation and the change-marking in different forms, depending on the implementation and user settings. The red background shown here is beneath the exact text (and layout) that was selected and then deleted in a single operation.

What's important is that the original list only has 3 items, not 4. The last line is not in the list. (The hanging paragraphs that are immediately below the list items numbered 2 and 3 are additional paragraphs of those list items.)
There is little doubt, given the deletion in Figure 1, that Figure 2 is the most straightforward and easily justifiable result. The model implemented in ODF blends the structure to the right of a deletion to fit the material retained to the left of the deletion. This appears to be the common practice in office-productivity software.

The problem that I have encountered is related to (1) rejection of the change failing and (2) the tracked-change being altered irreversibly when the document is reloaded, edited further in other places, and saved again.

The Original Text

![Image of the tracked deletion](CT-2011-04-02-2154-InitialList-ODF1.2-Lo3.3.2-dh.png)

**Figure 3: Original Text Before Any Deletion**

The original text is quite straightforward. The only unusual features are the introduction of an annotation under the list item numbered 3 and the use of additional hanging paragraphs in the list items numbered 2 and 3. (Tip: After pressing Enter after "list-item-b" in the second item, the automatically-supplied “3.” is removed by pressing backspace once in Libre Office 3.3.2, then typing the text of what will be an additional paragraph under item 2. The next enter will trigger “3.” again, etc. This convention works back at least as far as OO.o 2.4.1.)

The ODF 1.2 content.xml markup for the original text has the following abridged form:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<office:document-content
  xmlns:office="urn:oasis:names:tc:opendocument:xmlns:office:1.0"
  xmlns:text="urn:oasis:names:tc:opendocument:xmlns:text:1.0"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  office:version="1.2">
  <!-- This is CT-2011-04-02-2207-InitialList-ODF1.2-Lo3.3.2-dh.xml, extracted from CT-2011-04-02-2156-InitialList-ODF1.2-Lo3.3.2-dh.odt. Extraneous markup has been removed to narrow in on the portion of the document that is going to be used for a tracked deletion and its presentation. White space has been introduced to simplify review of the XML. -->
  <meta:generator>LibreOffice/3.3$Win32
```

The XML is easily matched to the document snippet in Figure 3.
Making the Deletion

The Tracked Deletion

1. This is list-item-a
2. This is list-item-b
   This is some text hanging below the list-item-b
3. This is list-item-c
   This is some text hanging below the list-item-c
This is a paragraph of text following the list.

Figure 4: Selection of the Material for Single Deletion

Although the selection begins in the middle of a list item and extends beyond the list into subsequent text, it is allowed by the document processor.

Confirming the Tracked Changes

When the Delete key is pressed in the selection of Figure 4, the change-tracking that is shown in the document presentation is that of Figure 1. Essentially, the exact selection area in Figure 4 is shown as deleted (light red background in the screen captures).

The following annotated extract of the resulting content.xml markup demonstrates how the tracking is implemented:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<office:body>
<!-- CT-2011-04-03-1257-ReTrackedDeletionRegression-ODF1.2-Lo3.3.2-dh.xml.
This is extracted from the content.xml part of the *.odt document of the same name. This <office:body> markup has been abbreviated to focus on the portions that involve deletion tracking.
White space has been added to simplify layout and review of the XML.
-->
<office:text text:use-soft-page-breaks="true">
  <text:tracked-changes>
    <text:changed-region xml:id="ct294068880" text:id="ct294068880">
      <text:deletion>
        <office:change-info>
          <dc:creator>Dennis Hamilton</dc:creator>
          <dc:date>2011-04-03T12:56:00</dc:date>
        </office:change-info>
        <!-- NOTE: See how the deletion provides synthetic
```

Note: Screen Capture CT-2011-04-02-2219-Selection-ODF1.2-Lo3.3.2-dh.png on opened document CT-2011-04-02-2156-InitialList-ODF1.2-Lo3.3.2-dh.odt.
start tags for those markup elements that are decapitated by the start-point of the deletion.

-->  
<text:list xml:id="list32690596" text:style-name="L1">
  <text:list-item>
    <text:p text:style-name="P1">list-item-b</text:p>
    <text:p text:style-name="P1">This is some text hanging below the list-item-b</text:p>
  </text:list-item>
  <!-- Here's a surprise interaction introduced in ODF 1.2 use of text:continue-list.  This points into the <text:deletion> element.
  -->
  <text:list-item>
    <text:p text:style-name="P1">This is list-item-c</text:p>
    <text:p text:style-name="P1">This is

<office:annotation>
  <dc:creator>Dennis Hamilton</dc:creator>
  <dc:date>2011-04-02T21:52:03.41</dc:date>
  <text:p text:style-name="P8">
    <text:span text:style-name="T8">This is some comment with some text</text:span>
  </text:p>
</office:annotation>

</text:list>
<text:p text:style-name="Text_20_body">This is a paragraph of</text:p>
</text:deletion>
</text:changed-region>
<!-- office:text markup preceding the heading above the deletion has been removed for simplicity of review. -->
<text:h text:style-name="Heading_20_1" text:outline-level="1">The Tracked Deletion</text:h>
<text:list xml:id="list32686892" text:continue-list="list32690596" text:style-name="L1">
  <!-- Here's a surprise interaction introduced in ODF 1.2 use of text:continue-list.  This points into the <text:deletion> element.
  -->
  <text:list-item>
    <text:p text:style-name="P1">This is list-item-a</text:p>
    <text:span text:style-name="T1">some text</text:span>

</text:list>
Stumble and Fall

At this point, everything should just work. But it didn't in the implementation I was using for my analysis.

As soon as the document with the deletion in Figure 1 was saved, the appearance of the change-tracked deletion changed. It was modified to look like this:
The Tracked Deletion

1. This is list-item-a
   
   2. This is list-item-b
      
      This is some text hanging below the list-item-b
   
   3. This is list-item-c
      
      This is some text hanging below the list-item-c
   
   4. This is a paragraph of text following the list.

Figure 5: The Tracked Deletion is Misinterpreted, Changing the List

Note: Screen Capture CT-2011-04-03-1300-AfterSaveReRegression-ODF1.2-Lo3.3.2-dh.png shows the corruption of the deletion after saving the document or reading the saved document.

For some reason, a list item numbered 4 is introduced into the deletion as shown in the change-marking of Figure 5. It appears that the implementation that presents the change-marking interprets the <text:delete> element as being in error, failing to detect that the </text:list-item> and </text:list> ending tags beyond the <text:change /> element are synthetic. The change tracking is now adjusted on the assumption that those tags were part of the original from which the deletion was made.

If the document is saved with this form shown, it can be seen that the <text:deletion> element has been modified to include an additional list item, with the </text:list> end tag moved to the end of the material.

As I write this, the content of section “The Tracked Deletion” looks exactly like Figure 5.

If I extract the content.xml from the ODF 1.2 package of this version of my working document, the corruption of the change-tracking is evident:

<?xml version="1.0" encoding="UTF-8"?>
<!-- CT-2011-04-04-1032-DeletionTrackingForensics-ODF1.2-Lo3.3.2-dh.xml extracted from content.xml of the *.odt of the same name. This extract shows only the <text:tracked-changes> element because it is what has changed. 
   NOTE: This is a few generations later after further editing and new saves (with new filenames) of the original. So many of the computer-generated ID and style-name values will have changed. 
   As usual, whitespace has been introduced to simplify presentation and review of the XML. 
-->

CT-2011-04-04-1123-DeletionTrackingForensics-ODF1.2-Lo3.3.2-dh dh:2011-04-04
Three Factors, Not Two, for Tracked Deletions

The Persistent Document Has But Two
We know there are two factors to a tracked deletion in the persistent document:

1. The post-deletion text that is correct as if the deletion is accepted

   This text will have synthesized end tags to ensure balancing of (a) those elements to the right that were beheaded by the deletion and (b) balancing of those elements that were truncated to the left of the deletion and that do not have natural mates to the right. In some cases, an element (a) will have its end tag altered to fit as a natural mate (</text:h> to/from </text:p>), although that does not apply in the case analyzed in this document.

2. The extracted text, which can be used to restore the text if the deletion is rejected

   Because deletions can tear the hierarchical XML structure of the before-deletion content.xml, they must be turned into a legitimate RNG schema text-content run of elements by affixing start tags and end tags to the torn-out fragment.

   It is the combination of the synthesis in (2) and the synthesis in (1) that must be unambiguously resolved and dealt with if the deletion is rescinded. In particular, any renaming and/or synthesis of end tags beyond the deletion point and removal of synthesized end tags in the <text:deletion> element must be synchronized properly.

The Presentation is Neither of Those
The third factor has to do with how an OpenDocument Consumer can present the tracked deletion for inspection, review, and perhaps further editing.

In general what is presented as change-marking is not directly-represented in the persistent document. That document has only the two factors identified above.

As we have seen, when tracked-changes are shown in the conventional manner, it is necessary to show the original structure but with the deleted material identified in some way. In effect, we should be presenting the pre-deletion form, as if the deletion were rejected, but with the excised material marked, just as in Figure 1 (and Figure 4 but with overlay of deletion marking rather than selection marking).

In this respect, the presentation is synthetic, just as it is when deletions and insertions are shown side-by-side. The presentation can be derived correctly from the change-tracking information, but it is not something that should be the basis for any alteration of the persistent document.
It should be clear that, for properly-auditable changes, it should not be possible to edit within the excision of an unrejected change, nor should it be possible to perform any alteration that crosses from part of the excision into the vicinity of the deletion seam.

I leave open the odd case where, in a replacement of text, it is possible in the usual implementation to reject the deletion but accept the replacement insertion.